



Publication number : **0 561 606 A1**

EUROPEAN PATENT APPLICATION

Application number : **93301972.1**

Int. Cl.⁵ : **H04N 1/21, H04N 1/32**

Date of filing : **16.03.93**

Priority : **20.03.92 US 855987**

Date of publication of application :
22.09.93 Bulletin 93/38

Designated Contracting States :
DE FR GB

Applicant : **XEROX CORPORATION**
Xerox Square
Rochester New York 14644 (US)

Inventor : **Cooper, Martin F. N.**
5093 Shalimar Circle
Fremont, California 94555 (US)
 Inventor : **Lo, Dick Wah**
1314 Chewpon Avenue
Milpitas, California 95035 (US)
 Inventor : **Johnson, Walter A. L.**
1345 Blackfield Drive
Santa Clara, California 95051 (US)
 Inventor : **Smith III, Z. Erol**
924 Los Robles
Palo Alto, California 94306 (US)

Representative : **Hill, Cecilia Ann**
Rank Xerox Patent Department, Albion House,
55-59 New Oxford Street
London WC1A 1BS (GB)

Method and system for labeling a document for storage, manipulation, and retrieval.

Storage, manipulation, and retrieval of files, for example data representations of scanned documents, is facilitated by establishing a relationship between an arbitrary, image domain file label (160) and a computer recognizable text domain file name for the file. Selection of the arbitrary, image domain file label is interpreted as a selection of the related file. The arbitrary, image domain file name is assigned by way of a paper form (150) or the like, and may be assigned at the time of document storage. The arbitrary, image domain file label facilitates the meaningful naming of a file for storage when a keyboard or other typical text entry apparatus is unavailable, such as when inputting a document for storage by way of a facsimile machine. Character recognition is not performed on the arbitrary, image domain file label, so the burden on the processing resource is minimized, while errors from inaccuracy are eliminated. Selection of a file for processing by way of its arbitrary, image domain file label may be by appropriate indication on either a screen display (Fig. 13) or a printed form (Fig. 12).

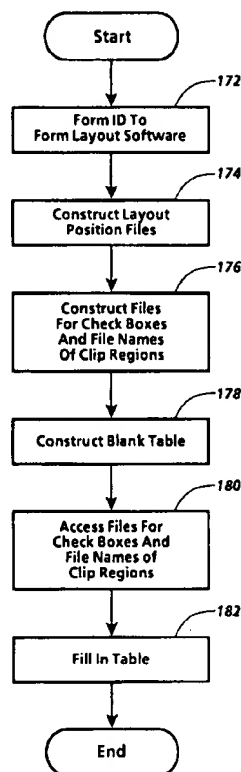


Fig. 11

EP 0 561 606 A1

The present invention relates generally to document processing methods and systems, and more specifically to a method and system for labeling a document with an arbitrary, image domain document label for document storage, manipulation, and retrieval.

Scanning documents for processing on a digital computer, such as a personal computer ("p.c."), a workstation, or other digital data processing resource is now routine. Furthermore, remote document storage, manipulation, and retrieval is becoming more commonplace today given the improving interfaces between computers and telecommunication devices such as fax machines. For example, a user can now "fax" a document to his computer for the purposes of storing the document on the computer, redistributing the document via the computer, etc. What ties these two different document processes together is that they both involve apparatus peripheral to the data processing resource. The present invention is concerned with facilitating the use of such peripheral apparatus, specifically the naming and referring to files stored on the data processing resource.

For purposes of the present discussion, the digital data processing resource such as the p.c., workstation, and the like will be referred to herein as a computer. Document as used herein shall be understood to mean a carrier, such as paper, for carrying markings, as well as the markings, if any, applied to the carrier. A file as used herein shall be understood to mean a collection of data, for example that representing a scanned image of a document, stored or accessible to a computer. The term electronic representation of data will be used herein, although the representation of the data (i.e., data representation) may be electronic, magnetic, optical, or other appropriate representation. Furthermore, the data may be in analog or digital format. Finally, document storage, manipulation, and retrieval will be understood to represent all actions that a user may perform on a document and its electronic representation, including those requiring communication between a peripheral apparatus and the computer. For example, this includes document scanning and transmission to the computer from a "remote" scanner, retrieving a file from the computer, transferring a document from one computer to another computer, etc. These definitions will simplify the explanation herein of the background and details of the present invention, although it will be understood that their use should not be interpreted as limiting the spirit and scope of the present invention.

Fundamentally, in order to perform any task on a document requiring communication between a peripheral apparatus and the computer, the document must be represented by data, i.e., an electronic representation of the document must be generated. Typically, the generation of an electronic representation of a document will be performed by a document scanner, which generates a description of the on/off state of the picture elements ("pixels") comprising the image, and packages the representation as a file. The form of the electronic representation may, for example, be a bitmap of the document or a coded collection of data representing the document.

Once an electronic representation of the document (hereafter referred to as an "electronic document") is generated, there must be a way of uniquely identifying it. This requirement is most commonly handled by the disk operating system resident on the computer. For convenience, virtually every disk operating systems permits, and in fact requires either the user or the computer to assign a file name to the file containing the electronic document for subsequent identification of the file. According to known document storage, manipulation and retrieval systems, the user-selected file name must be in a format which is recognizable by the computer, for example encoded text such as EBCDIC or ASCII which may be entered from a keyboard.

Electronic documents transmitted to a computer for storage and/or processing from a peripheral device are typically named at the time of transmission to or receipt by the computer in association with the task of document storage. For example, a user may enter via a keyboard attached to the sending or receiving device an encoded text name for the electronic document. Alternatively, the sending or receiving device may automatically assign an encoded text name to the electronic document according to a preestablished rule for name assignment. Typically, the task of document storage involves establishing a destination for the file in a memory media, such as a physical location on a magnetic disk, in RAM, etc., and a system identification ("system ID") of that destination. As part of the storage process, the disk operating system establishes and maintains a correspondence between the assigned file name and the system ID.

The file name, when assigned by the user, is often a mnemonic device or other label allowing a user to identify from the file name the general or specific contents of the file. When the file name is assigned by the system, it is most often a generic name such as, for example, the user's name, the name of the device from which the file was transmitted, the date and time of creation of the file, etc. Thus, a user is typically more likely to be able to identify the contents of a file when the user assigns the file name than when it is assigned by the system.

There are known systems that permit document retrieval using peripheral apparatus, such as a fax machine. One such system is disclosed in U.S. patent 4,893,333. According to this reference, a prestored document is identified for retrieval by way of indicia imparted on the form, for example, so-called bar codes, fill-in check boxes or fill-in fields. The idea of identifying a form absent such indicia by use of appropriate image

processing software is also disclosed therein. Furthermore, performing certain operations (store, retrieve, forward, etc.) on documents by way of a peripheral device, is provided when the document is capable of being identified by way of dual-tone DTMF telephone signals, as disclosed for example, in U.S. patent 4,918,722.

One problem continually encountered in the art is that not all peripheral devices are accompanied by a keyboard allowing the user to enter an appropriate file name, for example for assigning a file name for file storage, accessing prestored files, etc. A typical stand-alone scanner comprises optical imaging components, software for processing images, and possibly paper document handling mechanisms. Typical facsimile devices include the above as well as a numerical keypad, but rarely include all of the keys of a full alpha-numeric keyboard. In general, present peripheral apparatus limit the ability of the user to assign a meaningful file name to files and access previously stored files.

Furthermore, when identifying pre-stored and pre-named files by way of filling in check boxes or fill-in fields, at least one check box or fill-in field must be appropriately marked for each character in the file name. This leads to time consuming and error prone document identification. For example, if check boxes are employed to identify a file, a great many such check boxes must be provided to allow identification of alphanumeric file names. If fill-in fields are employed, the processing apparatus which identifies the document must ultimately perform character recognition on the indications in the fill-in fields.

Finally, virtually every system for establishing file names requires not only that the file name be in a format which is recognizable by the computer, but that the character set used in the file name be the native character set of the computer. For example, it is generally not possible assign a file name to a file using a foreign language character set or graphics unless the processing apparatus is capable of recognizing the character set or graphics. This precludes such operations as assigning a file a file name with Kanji characters when the computer is capable of recognizing only the Latin characters et.

The present invention provides an image processing system comprising processor means; memory means, connected to said processor means, containing a data item; conversion means, connected to said processor means, for converting a printed document into a file of data representing the printed document; image processing means connected to said conversion means for extracting from the file data representing selected indications imparted on the printed document; and means for establishing a relationship between the data representing selected indications imparted on the printed document and the data item such that selection of the data representing selected indications imparted on the printed document will be interpreted as selection of the data item. The conversion means of a system in accordance with the present invention may be a facsimile machine.

The selected indications imparted on the printed document may include at least an arbitrary, image domain file label. In the case wherein the data item includes a data representation of a second document, the arbitrary, image domain file label may be a document label for the second document. The image domain file label may be a user imparted handwritten file label.

In accordance with another aspect of the present invention, there is provided a document storage manipulation and retrieval system, comprising processor means; memory means connected to the processor means such that the processor means may store and retrieve electronic data therefrom; document imaging means connected to the processor means, for acquiring an image of a printed document and converting the acquired image into a data representation of the document; image processing means connected to said document imaging means for identifying from the data representation of the document at least one selected region of the document; software product means communicably connected to said processor means and said image processing means for assigning a first file name to a data representation of a first document acquired by said document imaging means, for assigning a second file name to the selected region of a second document identified by the image processing means from the data representation of the second document, and for establishing a relationship between the first and second file names such that identification of the contents of the second file may be interpreted by the processor means as identification of the contents of the first file.

The selected region of the second document identified by the image processing means may contain an arbitrary, image domain document label. The image domain document label may be a user imparted handwritten document label.

The present invention further provides a data storage medium communicably interconnected with a digital data processing resource, said data storage medium having stored thereon a data structure defining a relationship between data representing a file and data representing an arbitrary, image domain file label such that a selection of the data representing an arbitrary, image domain file label will be interpreted by the data processing resource as a selection of the file. In the case wherein the first file contains a data representation of a document, the arbitrary, image domain file label may be a document label. The image domain document label may be a user imparted handwritten document label.

The present invention also provides a method for assigning an arbitrary, image domain file label to a first

file having a first file name stored in the memory of a digital data processing resource, comprising the steps of converting a first document containing at least one field for receiving an arbitrary, image domain file label to a data representation thereof; determining the location of said at least one field in the data representation of the first document; assigning a second file name to the data representation of the contents of the at least one field; and establishing a relationship between the first and second file names such that a selection of the second file name will be interpreted by the digital data processing resource as a selection of the first file name. In the case wherein the digital data processing resource utilizes a data base for at least a part of its data processing tasks, the relationship between the first and second file names may be a link in the data base.

In accordance with another aspect of the present invention, there is provided a method for operating a document processing system in which first and second files are stored, and further in which a relationship has been established between the contents of the first and second files such that selection of the contents of the first file is interpreted by the document processing system as a selection of the contents of the second file, comprising the steps of designating a processing task to be performed on a document; selecting the contents of the first file for the processing task; and obtaining, by way of selection of the contents of the first file, the processing task being performed on the contents of the second file. The contents of the first file may be data representing an arbitrary, image domain file label, in which the case the method may further include the steps of displaying the arbitrary, image domain file label contents of the first file for selection, and selecting the arbitrary, image domain file label so as to select the contents of the first file for the processing task.

The present invention further provides a method for operating a data processing resource, comprising the steps of inputting a first paper document into a facsimile apparatus; causing the facsimile apparatus to convert an image of the first paper document into an electronic data representation thereof, and transmitting the electronic data representation of the first paper document to the data processing resource; storing in a first file the electronic data representation of the first paper document in a memory apparatus connected to the data processing resource such that a first file name is assigned thereto; inputting a second paper document having at least a user-imparted arbitrary, image domain document label imparted thereon into the facsimile apparatus; causing the facsimile apparatus to convert an image of the second paper document, including the user-imparted arbitrary, image domain document label, into an electronic data representation thereof, and transmitting the electronic data representation of the second paper document to the data processing resource; storing in a second file the electronic data representation of the second paper document in a memory apparatus connected to the data processing resource such that a second file name is assigned thereto; determining the location of the electronic data representation of the user-imparted arbitrary, image domain document label in the electronic data representation of the second paper document; storing in a third file the determined location of the electronic data representation of the user-imparted arbitrary, image domain document label such that a third file name is assigned thereto; and establishing a relationship using links provided by a data base between the first and third files such that selection of the contents of the third file will be interpreted by the data processing resource as a selection of the contents of the first file.

The present invention overcomes the problem of the limits imposed on a user in entering a file name by providing a method and system for assigning a meaningful user-selected file label to files which uses existing peripheral devices. Specifically, for a file having an assigned file name, a relationship is established between an image domain file label and the file name assigned by the computer, so that the label may be employed to assist the user in identifying the file.

The present invention builds on the methods and systems of the prior art by providing a relationship between the assigned file name and the image domain file label for a file. This allows establishing a meaningful file label for a file which can stand in the place of the less meaningful assigned file name. Furthermore, by establishing this relationship, a user may more easily and directly identify a desired file in a system lacking a text entry device than heretofore provided by the prior art.

One aspect of the present invention involves storing a file on or by way of a computer. According to this aspect, the file is initially a document consisting of a carrier means such as paper, plastic, etc., having markings such as printing or writing thereon. A special cover form is employed which includes a region in which the user imparts an image domain label (for example a handwritten name or illustration) for the file. The document, prefaced by the cover form, is scanned by a scanning means whose output is an electronic data file containing the image of the form and the document. This data file is transmitted to a computer, where it is assigned a file name and stored as a file either in the computer's memory or in a memory media associated with the computer.

Associated with the transmission of the data file to the computer will be an instruction to the computer to store the file (the instruction being read from the form or other input device). The computer establishes a location in which to store the file and creates a file name for the file. The computer maintains the association between the location of the file and the file name according to methods well known in the art. Next, the computer distinguishes the data representing the form and the data representing the document, locates represen-

tation of the image domain label imparted on the form, and establishes a relationship by way of data base entries between the data representing the image domain label and the data representing the document. When the computer is called on to access the document, it displays or prints the image domain label in such a manner that selection of the image domain label is interpreted by the computer to mean selection of the document.

Another aspect of the present invention involves accessing for sending, retrieving, deleting, etc., a previously stored electronic document having related to it an image domain file label. According to this aspect, a user would request a listing of the labels of an appropriate set of files which are stored on or accessible to the computer. In response to the request for the listing, the computer generates a display of the image domain file label, if any, and possibly other indications, for each file. The display may be an image formed on a computer display, a printed paper document, etc. From this display, the user selects the item(s) of interest by selecting the image domain file label, for example by highlighting the file label on the computer display or imparting a check mark in a check box field on a paper or other printed document of the display. Based on the preestablished relationship between the image domain file label and the file name, the computer is able to interpret the user's selection as a selection of the associated file.

Closely related to the above is the aspect that an image domain label may be assigned to a file and used to identify that file without resort to character recognition software such as optical character recognition ("OCR"). That is, there is no requirement to convert the image domain file label into a machine recognizable format. This reduces the demands on the processing resources of the computer, increases the speed at which the computer can process instructions involving the image domain label, allows use of characters other than those supported by the character set of the computer (e.g., Kanji characters used on a standard DOS machine), allows use of non-textual labels (such as figures or relevant non-textual marks), and allows the user to select the image domain label without requiring the user to duplicate that label.

Yet another aspect of the present invention is that the file to which the image domain label is assigned need not be an electronic document. For example, the file may be data representing one or more instructions, or a program of instructions, which the computer will follow to accomplish specific tasks. That is, the underlying subject matter having the associated image domain label may be a computer program which may be referred to, loaded, and/or run in or by the computer by referring to the image domain label. Alternatively, the file to which the image domain label is related may be one or more of many other types of files, such as binary files in formats utilized by other data processing resources. In fact, the term "file" is used herein in its broadest sense to refer to the element to which the image domain label is assigned, and shall be understood to mean any data item or portion of a data item which is appropriate for assignment of an image domain label.

By way of example only, embodiments of the invention will be described with reference to the accompanying drawings. In the drawings, like reference numerals will be used to refer to like elements as between the various figures, in which:

Fig. 1 shows an apparatus, including a computer and peripheral devices of the type which might typically employ or be a part of an embodiment of the present invention;

Fig. 2 is an illustration of various software modules, including a common communicational interconnection between same, of the type which might typically employ or be a part of an embodiment of the present invention;

Fig. 3 shows an apparatus including a computer and a facsimile machine capable of both sending data to and receiving data from the computer of the type which might typically employ or be a part of an embodiment of the present invention;

Fig. 4 is an illustration of a batch, showing jobs, documents, and forms;

Fig. 5 is a flow diagram illustrating the steps to accomplish remote document storage in one embodiment of the present invention;

Fig. 6 is a flow diagram illustrating the steps involved in transferring a batch to the computer for processing in one embodiment of the present invention;

Fig. 7 is a flow diagram illustrating the steps involved in processing a batch in one embodiment of the present invention;

Fig. 8 is a flow diagram illustrating the steps involved in a job set action in one embodiment of the present invention;

Fig. 9 is an illustration of a form utilized in one embodiment of the present invention;

Fig. 10 is an illustration of an action table generated by image processing software in one embodiment of the present invention;

Fig. 11 is an illustration of the steps involved in providing a table for the establishment of processing actions in the job data base in one embodiment of the present invention;

Fig. 12 is an illustration of a selection form relating to the send action generated in response to the request for a list of documents for sending in one embodiment of the present invention; and

Fig. 13 is an illustration of a computer display screen displaying a list of image labels in a windowing user interface environment to facilitate document selection using an input device such as a mouse.

Referring to Fig. 1, there is shown an apparatus 10 which includes a computer 12 (such as a p.c., workstation, server, or other digital data processing resource) to which one or more peripheral devices may be communicably interconnected. These peripheral devices include devices designed primarily for communicating to computer 12, such as a network connection 14, a form editor 16, a scanner 18, an image communication means 20 (such as a facsimile or "fax" machine), and a storage device 22 (such as a magnetic, optical or electrical storage device), and devices designed primarily for receiving communication from computer 12, such as a network connection 24, a display device 26 (such as a CRT), a printer 28, an image communications means 30 (such as a fax machine), and a storage device 32 (such as a magnetic, optical or electrical storage device).

Computer 12 will generally include a central processing unit (not shown) which performs processing of data under control of various software modules. With reference to Fig. 2, these modules include, *inter alia*, disk operating software module 40, a user interface software module 42, a fax interface software module 44, a data base software module 46, applications software module 48, and optionally, a terminate and stay resident ("TSR") software module 50 whose instructions may be loaded from either or both of the fax interface software module 44 and the applications software module 48. One or more of the various modules may form a software product. The various modules would typically be in communication with one another roughly as illustrated. In particular, in one embodiment of the present invention, applications software module will include image processing software module 48a, data base instruction software module 48b, interface and I/O software module 48c, form description language software module 48d, form layout software 48e, and other modules for accessing and processing data as will be understood by one skilled in the art.

In order to clearly illustrate one embodiment of the present invention, an apparatus 52 is illustrated in Fig. 3 which includes a computer 54, and a single peripheral device, which in this embodiment is a fax machine 56 capable of both sending data to computer 54 and receiving data from computer 54 via common telephone lines 58. Furthermore, computer 54 will be assumed to include, *inter alia*, a display device 54a, an input device 54b, a fax card interface module 54c (including fax interface software module 44 (Fig. 2), and any additional hardware and software for enabling the computer to receive and send fax data via telephone lines 58), and processing and memory unit 54d.

The purpose of apparatus 52 is, at least in part, to allow a user to convert a paper document to an electronic document via fax machine 56, send the electronic document over telephone lines 58 to computer 54 via standard facsimile communication protocols, such as CCITT group 3, to command computer 54 to perform certain operations (hereafter referred to as "tasks") by way of marks made on paper which are converted and sent to computer 54 per the above, and to print documents at fax machine 56. In this way, fax machine 56 serves to perform the tasks of 3 separate devices - (a) an input scanner, (b) a computer operator's interface, and (c) a printer.

The operation of apparatus 52 of Fig. 3 will now be described. Assume that a user is at a physically distant location from computer 54, but that fax machine 56 is located at the user's distant location. Suppose further that a user has a paper document which the user desires to store in electronic format on computer 54. This task will be referred to as remote document storage (or more succinctly as the "store" task). The paper document may be of virtually any type, for example one having text and/or illustrations imparted thereon. The steps in accomplishing this task will be discussed with reference to Figs. 4 and 5.

With reference to Fig. 4, the user will assemble together one or more pages which comprise a document 60, and preface document 60 by an instruction form 62. Instruction form 62 will generally be a single page form, although multiple page instruction forms may be appropriate in certain circumstances. Together, document 60 and the instruction form 62 are referred to as job 64. It will be appreciated that a user may input to computer 54 (Fig. 3) one job at a time, or may submit plural jobs 64, 64', etc., together. As a collection, the job or jobs assembled for transmitting to computer 54 is referred to as a batch 66.

Once assembled (in the order shown from left to right in Fig. 4), batch 66 is loaded into fax machine 56 (Fig. 3) where its image is captured and an electronic form of the image of the batch generated (hereafter referred to as "batch image data"). Fax machine 56 will generate the batch image data for transmission to computer 54 in a standard format, such as the CCITT group 3 encoding format.

Figs. 5 through 8 are flowcharts illustrating the steps for performing remote document storage on computer 54. Fig. 5 shows the initial steps performed by fax machine 56 to accomplish remote document storage. To begin, fax machine 56 dials the number of the fax interface module 54c, which is provided to it by the user, as shown at step 100. Fax module 54c is then polled at step 102 to determine if it is ready to receive the batch image data. If the fax interface module 54c is not ready to receive the batch image data, the fax machine may disconnect the line and retry dialing the fax interface module at a later time. However, if the fax interface module 54c is ready to receive the batch image data, the batch image data is generated by the scanning and processing

hardware and software of fax machine 56, as shown at step 104. The batch image data is then transmitted via telephone line 58 to the fax interface module 54c at step 106. The batch image data is received by the fax interface module 54c, which has resident memory or utilizes a portion of the memory of processing and memory unit 54d for temporarily storing the batch image data. Fax interface module 54c will automatically assign an appropriate file name and/or system ID to the batch image data file. Computer 54 then processes the batch image data as follows.

With reference to Fig. 6 (which is a flow chart illustrating the steps in transferring a batch to the computer for processing) the fax interface module 54c is polled, periodically, to determine if it has received batch image data, as shown at step 108. Since processing begins at the batch level, step 110 is to determine whether a complete batch has been received. One method for doing so is to determine whether the telephone connection between fax interface module 54c and fax machine 56 has been broken. If so, it may be assumed that the received batch image data is a complete batch. If it is determined that a complete batch has been received, the batch must next be transferred to the computer.

For the purpose of the following description it will be assumed that TSR software module 50 acts as an interface between fax card interface software module 44 and applications software module 48. This is a convenience which facilitates processing, and represents only one of many ways to establish such an interface.

Furthermore, a data base called a job data base may be used as a scheduler to control the performing of certain tasks by the computer. The job data base is comprised of defined entries called actions, each action having a link to other actions and/or to entries in a second data base referred to as an information data base, which functions primarily as a repository for data used by, *inter alia*, the job data base. An action entry will include data indicating the action's function, and data that can be used in scheduling performance or execution of the action's function. Each task will have at least one action associated with it. A list of possible actions with their definitions is given in Appendix 1, attached hereto. It will be appreciated that the scheduling of actions and organization of stored items may be handled by traditional methods involving a CPU, main memory, etc. as well known in the art.

The next step is to transfer the batch to computer 54 for processing. A file is created at step 112 (hereinafter referred to as an "event file") for maintaining relevant information about the batch. The event file is assigned a name automatically by the computer, for example of the type FAXAAAA.EVT, where AAAA represents a four digit integer. This may be handled, for example, by TSR software module 50, by maintaining and/or referring to a portion of the computer's memory reserved for keeping track of the value of the last integer assigned to a file name. In this way, each event file gets a sequentially numbered file name.

TSR software module 50 will then cause certain information to be written into the event file including, for example, that there is a new batch image data file stored, and the file name assigned by the fax interface module 54c to that batch image data file. This facilitates processing of the batch by the applications software module 48 as follows.

Applications software module 48 will periodically poll the TSR software module 50 to determine if there is a new batch image data file for processing, as shown at step 114 of Fig. 6. One convenient way of accomplishing this is to ask TSR for the current event file number. If TSR has no new event file, the current event file number would be set to 0. Thus, the applications software would interpret a 0 in response to its poll as an indication of no new event file. However, if TSR has a new event file to pass to the applications software module 48, TSR would respond to a poll with an integer representing the integer assigned to the event file as described above. Thus, applications software module 48 would interpret receipt of a non-zero integer, for example BBBB, in response to its polling by forming the file name FAXBBBB.EVT. This is represented by the receipt of file name for the event file at step 116. It would then look for the information stored in file FAXBBBB.EVT for further processing. This is handled by an action called an input action. In this way, access to the batch image data file is quickly and simply facilitated.

At this point, the batch image data file is a DOS file in a standard encoded fax format, and may be stored in a file format particular to fax interface module 54c. In order to facilitate a uniform processing of the batch image data, the applications software module calls a conversion function which, using the information in the event file, converts the batch image data into an appropriate intermediate data format. This step is shown at 118 of Fig. 6.

Processing of the batch will now be described with regard to Fig. 7. In the aforementioned intermediate data format, the batch is treated as a collection of discrete pages; the standard fax format which fax machine 56 (Fig. 3) created the batch image data in will contain indications, such as start or end of page indications, allowing page differentiation. This facilitates the first step in processing the job, execution of the batch action, which creates a data base entry in the job data base for each page of the batch image data at step 120. This may be accomplished by examining the batch image data for the aforementioned page indicators, or by transferring from the fax interface module 54c a count of the number of pages in the batch, for example by way of

the event file. This enables appropriate image processing software module 48a to examine each page of the batch individually, and to record for each page appropriate data in a separate page entry. For example, such data may include whether a page is an instruction form 62 or not, which batch the page is a part of, etc.

Execution of the batch action will cause a page action to be associated with each page entry in the information data base. These page actions causes image processing software module 48a to examine each page in the batch image data to determine whether the page is a form or not, and record the form/not form information in the page entry in the information data base. This is shown at steps 122 to 128.

Returning to Fig. 4 for the moment, each instruction form 62 includes a form data region 68 which can carry coded data of various types. In one embodiment of the present invention, coded form data region 68 contains a code which allows computer 54 to identify the form and the steps required to process the form and associated document(s), if any. Alternatively, coded data region 68 may contain a complete description of the form and how to process the form and associated document(s) as described in United States Patent No. 5,060,980. The format of the coding may be of the type described in European Patent Application No. 0 469 864 or may be another computer readable coding scheme as appropriate.

By examining form data region 68, the applications software module may determine whether the subject page is a form or not. The presence of coded data in form data region 68 indicates that the page is a form, while the absence of coded data indicates the page is not a form.

An alternative method and device for determining whether a subject page is a form or not is to include on form pages a logo or monogram in region 70, and to employ appropriate image processing software to determine whether a page contains the logo or monogram in that region. Again, presence of the logo or monogram in region 70 indicates that the page is a form, while absence of the logo or monogram indicates that the page is not a form.

If a page is determined to be a form, image processing software module 48a next examines the form data region 68, and identifies the form. Typically, the information encoded in region 68 will be a form identifier, pointing to a form description, and steps for processing the form and any associated document(s), stored in the information data base or in the computer's memory. The form identity is then also stored in the page entry for the form page in the information data base. Returning to Fig. 7, this processing is shown at steps 132 and 134.

When each of the page actions are completed, and no more pages remain in the batch to be examined, a job set action is executed, which is also established by the batch action. The job set action will be described with regard to Fig. 8. The function of the job set action is to break the batch up into discrete jobs comprising a form and associated document(s), if any. The page entries for the batch are examined at step 138. The determination of whether a page is a form or not is made at step 140. If a page entry is indicated to be a form, a new entry is created in the job data base, called a job entry, at step 142. The page entry containing a form indication is then entered into this job entry at step 144. If the page entry does not contain an indication that the page is a form, the page entry is entered into the current job entry. A determination is then made at step 146 as to whether the page entry just processed was the last page entry for the batch. If not, the next page is examined at step 138 per the above. If so, the end of the batch is reached, the batch has been fully divided into jobs, and the job set action is complete.

Once each job is defined, processing of the individual jobs may proceed. Initially, since each job begins with a form, a form action is created for each job. The role of the form action is to assemble the form description and any other pertinent data required by the image processing software to set the computer up to process the electronic document. At this point, a brief description of a form, such as form 150 shown in Fig. 9, will assist in an understanding of the processing of the job. Form 150 will be divided into one or more distinct regions, for example header region 152, store region 154, retrieve region 156, list region 158, etc. Each region may carry markings such as writing, coded information, or illustration, and user-modifiable fields such as clip region 160 or check boxes 162. Other such fields are within the scope of the present invention as well.

As part of the form action, the form identification is used to access the form description in the information data base. Specifically, a data structure is created which contains a description of the location of the various user modifiable fields located on the form (if any). The form description and the batch image data file name are then presented to image processing software, which examines the user-modifiable regions for user modifications. That is, if the form is defined to have a clip region at a particular location, the image processing software locates that clip region on the form and clips the contents of (i.e., extracts the image from) that region. Likewise, if a form has a check box defined to be located at a particular location, the image processing software determines whether the box has been checked or not.

For convenience, the output of the image processing software analysis of the form is in the form of entries in a table established by the applications software module as follows. An example of a table 170 is shown in Fig. 10. Table 170 will have a number of entries, one or more of which correspond to a task which the form

will cause the computer to perform. The steps for filling in table 170 are illustrated in Fig. 11. Initially, the form identification will be made available to form layout software module 48e as shown at step 172. From the knowledge of the form identification, form layout software module 48e is capable of constructing files whose contents indicate the positions of the various check boxes and clip regions, as shown at step 174. This indication might, for example, be in x-y coordinate values from a convenient reference point on the form.

The position files are then passed to the image processing software module 48a along with the page entry corresponding to the form page. The image processing software then locates the check boxes, if any, and determines for each check box whether the box is checked or not. This information is entered into a file. The image processing software also locates the clip regions, if any, and for each clip region clips the contents of the region and places the contents in a file. The file names for the files containing the contents of the clip regions are then put into a file. Together, these processes are shown at step 176.

From the form identification, the form description language software module may construct a blank table corresponding to that form at step 178. The files containing the information about which check boxes are checked and the file names of the files containing the contents of the clip regions are then accessed at step 180, and the data these files contain are used to fill in the blank table, resulting in a complete table 170 (Fig. 10) at step 182.

Returning to Fig. 10, table 170 will be divided into 4 columns. A first column 184 will be reserved for a pre-action step. The entries in this column will be used to determine which action entries will be made in the job data base to accomplish the task. For example, if the task is retrieve, the entry in column 184 will cause pre-retrieve and retrieve actions to be entered into the job data base. A second column 186 will be reserved for a modifier for the pre-action step. The entries in this column further determine what the action will be entered in the job data base. For example, if the task is to produce a list, the entry in column 186 will indicate whether the list is a retrieve list, send list, delete list, etc. A third column 188 will be reserved for a state indication. The entries in this column indicate whether a corresponding check box has been determined to be checked or not. Finally, a fourth column 190 will be reserved for a parameter indication. The entries in this column are, for example, the name of the file containing the clip region image forming the image domain file label.

Returning to Fig. 11, once access to table 170 has been provided to the applications software module, processing of the job proceeds by establishing appropriate actions in the job data base to execute the actions indicated in the pre-action step column 174, and to facilitate their execution by providing the items called for in the parameter column 180. In order to do so, the table is circulated through a sufficient number of times such that a sequential execution of the actions in the job data base will cause the desired result. For example, for the store task, an association between the file name of the document to be stored and the file name for the arbitrary, image domain document label must be established and maintained. The document itself is all pages of the job except for the form. Thus, the page entries in the job data base for the job are examined, and those entries which indicate that their associated page is not a form are entered into a document entry in the information data base. The file name of the arbitrary, image domain document label entered into the parameter column 190 of table 170 is linked with the pages of the document by way of an entry into the document entry. In this way, a relationship is established between the electronic document and the arbitrary, image domain document label.

It should be noted that this relationship may be subsequently altered by the user as convenient. For example, the user may have resorted to the image domain label because a keyboard was unavailable. However, the user may desire to convert the image domain label to a format which is recognizable by the computer, for example encoded text such as EBCDIC or ASCII which may be entered from a keyboard. This may be accomplished by establishing a function, in the manner of renaming a file, which permits the deletion of the image domain file name and the substitution of a text domain file name. Substituting an image domain file name for a text domain file name may be achieved by a similar operation in association with, for example, a scanner.

At this point, a user has stored the document on the remotely located computer, and a relationship has been established between the document and the document label. As will be shown, this will facilitate the user's performing many operations on the document, including retrieving the document, sending the document to another party, deleting the document, etc. For convenience, the following discussion focuses on sending the document. Since it will be understood that the description readily extends to many tasks that may be performed on, by or with the stored document, these other tasks are only discussed where their performance requires a substantial deviation from the description.

Initially, it will be assumed that the user is again at a location remote from the computer. In order to implement the send task, the user first requests from the computer a list of stored documents. In one embodiment of the present invention, this request is made by way of a paper form, such as form 150 of Fig. 9. The form may be a dedicated list request form, or may be a multi purpose form, in which case the user may be required to select the list task and sections in which the list will be presented, such as send, retrieve, delete, etc.

This form is sent to computer 54, where it is received by fax interface module 54c, while the input action is executed, etc., as detailed above.

The form requesting a list of documents will then be determined to be one invoking the list task, and processing will proceed to determine what actions must be invoked in order to accomplish the requested task. In response to the list task, a form will be generated having a list of possible recipients and a list of the documents which may be sent, and check boxes associated with each such that selection of a check box will ultimately be interpreted by computer 54 as a selection of the recipient or document associated with that check box. This form will then be transmitted to the user, for example by the computer dialing the user's fax machine and transmitting the form to the user's fax machine for printing.

The user will then indicate (i.e., select) on the send form the recipient(s) for the document(s) and the document(s) to be sent by placing an appropriate mark in the check boxes corresponding to each. This marked form is then sent to computer 54, where it is received by fax interface module 54c, while the input action is executed, etc., as detailed above. Upon processing the send form, a pre-send action is entered into the job data base which converts the indicated documents into the appropriate format for fax interface module 54c, and enters a send action for each recipient indicated on the form. Each send action then queues all indicated converted documents and instructs the TSR software module to coordinate the sending of each document, whether it be by way of fax transmission, network communication, or otherwise.

Retrieving a form will proceed in much the same fashion, with pre-retrieve and retrieve actions established and appropriate documents indicated. (It will be appreciated that by designating as a recipient the remote location that the user is at, it is possible to retrieve a document as an alternative to the retrieve task.)

One important feature of the send list generated by the list task is that it will present to the user the image domain label which was related to the document as detailed above. Again, the label may be of virtually any appropriate marking, and several examples are shown in Fig. 12. For example, labels 200 and 202 are handwritten character-based image labels, label 204 is a typewritten character-based image label (for example, in a text type recognizable by the computer), label 206 is an illustration-based image label, and label 208 is a non English language character based image label.

An alternative to invoking the send, retrieve, etc., actions by way of a form is to do so on a display, such as a CRT, of a computer. For example, suppose that the user has now returned from the distant location and has access to computer 54 on which has been stored one or more documents according to the above technique. The application software will facilitate obtaining a list of all or selected sets of the stored documents. A display screen 210 showing such a document list 212 in a windowing environment is illustrated in Fig. 13.

List 212 will include, for each listed item, an item type icon 214 (for example, a document icon, etc.), an image domain label 216, and other pertinent data 218 (for example, file size, creation date, etc.) when appropriate. As a precursor to constructing the document list 212, a table is constructed for managing the various file names, although this table is not displayed. When the document list 212 is initially requested, the information data base is examined to determine what files have been stored. The document entries of the information data base are examined, and the document file name and the file name of the related image label are entered into the table. This table is used to construct list 212 such that the file containing the image label is displayed, with a link to the related document. That is, at each row in the list an image label is displayed. A user's selection of a row will be interpreted to mean a selection of the document related to the image label displayed in that row.

The user interface will have a particular protocol for identification (i.e., selection) of a displayed item. For example, clicking a button on a mouse input device is commonly interpreted as a selection of the region at which the pointer on the display screen is pointing. By way of standard interfacing with the user interface, it is possible to define what function selection of a particular displayed item will result in. For example, selection by double-clicking a mouse button is commonly defined as causing the selected file to be opened and displayed.

The applications software does not necessarily know whether the form it receives comes from a scanner, a fax machine, another computer, etc. Thus, one variation of the above involves providing a document to the computer for storing, sending, etc., from another computer. The document, which would be stored as a file on one computer, could be prefaced by an electronic form which the destination computer would interpret just as if the form and document were originally in paper form and were faxed to it as described above. One consequence of storing a document in this regime is that the document label will appear in the font of the computer on which the form was generated. However, the document label will be treated just as any other document label, and will not be stored in a computer recognizable form.

In general, to those skilled in the art to which this invention relates, many changes in construction and widely differing embodiments and applications of the present invention will suggest themselves. For example, the present invention has been described in terms of remote document storage, manipulation and retrieval. However, it will be appreciated that the foregoing applies not only to documents, but to other types of data as

well. In addition, the foregoing has been from the perspective of assigning a name to a file. It will be appreciated that the same procedures would apply to establishing any other relationship between a reference item and an item to be referred to, for example a recipient's name and a recipient's telephone number however stored.

5

10

15

20

25

30

35

40

45

50

55

APPENDIX 1

5 When selecting the next action to be performed, the job scheduler may select one of 16 different types of action:

10 1. Input

This is the default action, and is executed when no other action is ready. It polls the application TSR to determine whether or not there is a newly received fax to be processed, and if so, enters a new batch in the database, and a new Batch action.

15

2. Batch

For a normal (image) fax transmission, this action creates, for each incoming page, a page entry in the database along with a Page action to be performed on that page, and a Job Set action. For a binary fax transmission, a job is created along with a Store Binary action. In both cases, a Cleanup action is also created.

20

25 3. Page

This action calls the image processing code to determine whether the page is blank (only if there is a single page in the batch), is a form, or is a data page. This determination is then stored in the database. For a form, the coded form data is also retrieved from the form image and stored in the database.

30

4. Job Set

This action is determined to be ready when there are no pages in the batch whose type remains unknown (i.e. when all corresponding Page actions are complete). Based on the sequence of forms and data pages, jobs are created in the database along with either a Form action (if the job has a form) or a Data action (if it does not) for each job.

40

5. Data

This action extracts the appropriate pages from the incoming fax transmission and creates an entry in the Information Database which corresponds to incoming fax mail.

45

6. Form

This action uses the coded form data from the form to find the form description file, then processes the form description to build data structures which are then passed to the image processing code. The image processing code determines which check boxes have been marked, and extracts any clip regions which have been used (e.g. a cover note). The results are then

55

used to interpret the instructions on the form and create various actions and other data structures within the database.

5

7. Pre-Send

This action converts any files requested for send to the file format native to the fax card in use on this particular system. It also processes all the requested recipients and creates a Send action for each of those.

10

8. Send

This action queues all the requested files and submits a request to the application TSR to send them to the specified recipient. It also creates a corresponding Verify action.

15

9. Store

This action extracts any data pages associated with the job and creates a corresponding entry in the Information Database. It also adds the new virtual file to any categories which were specified on the form.

20

25

10. Store Binary

This action creates an entry in the Information Database for an incoming binary file, and marks this as incoming fax mail.

30

11. Pre-Retrieve

This action converts any files requested for retrieve to the file format native to the fax card in use on this particular system. It also creates a Retrieve action.

35

12. Retrieve

This action queues all the requested files and submits a request to the application TSR to send them to the return address. It also creates a corresponding Verify action.

40

13. Delete

This action deletes any files (from both the Information Database and the disk) for which deletion was indicated on the form.

45

14. Verify

This action requests the status of its corresponding Send or Retrieve event from the application TSR, in order to verify its completion or failure. If the event has not yet completed, the action resets its start time so that it will verify later.

50

55

15. Cleanup

This action runs after all other actions associated with a batch are complete. It tidies up after all the other actions by deleting any temporary files.

16. Purge

This action is independent of any batch. It runs periodically, as specified by the user, and deletes old unwanted information from the database.

Claims

1. An image processing system, comprising:
 - processor means;
 - memory means, connected to said processor means, containing a data item;
 - conversion means, connected to said processor means, for converting a printed document into a file of data representing the printed document;
 - image processing means connected to said conversion means for extracting, from the file, data representing selected indications imparted on the printed document; and
 - means for establishing a relationship between the data representing selected indications imparted on the printed document and the data item such that selection of the data representing selected indications imparted on the printed document will be interpreted as selection of the data item.
2. An image processing system as claimed in claim 1, further including listing means connected to the processor means for generating, upon request, a listing of an image of the indications represented by the data representing selected indications imparted on the printed document, such that the established relationship between the data representing selected indications imparted on the printed document and the data item facilitates the selection of the image from the listing being interpreted by the processor means as a selection of the data item.
3. An image processing system as claimed in claim 2, wherein the list generated by the listing means generates a printed listing of an image of the indications represented by the data representing selected indications imparted on the printed document, such that the established relationship between the data representing selected indications imparted on the printed document and the data item facilitates the selection of the image from the listing being interpreted by the processor means as a selection of the data item.
4. A document storage, manipulation and retrieval system, comprising:
 - processor means;
 - memory means connected to the processor means such that the processor means may store and retrieve electronic data therefrom;
 - document imaging means connected to the processor means, for acquiring an image of a printed document and converting the acquired image into a data representation of the document;
 - image processing means connected to said document imaging means for identifying from the data representation of the document at least one selected region of the document;
 - software product means communicably connected to said processor means and said image processing means for assigning a first file name to a data representation of a first document acquired by said document imaging means, for assigning a second file name to the selected region of a second document identified by the image processing means from the data representation of the second document, and for establishing a relationship between the first and second file names such that identification of the contents of the second file may be interpreted by the processor means as identification of the contents of the first file.
5. A system as claimed in claim 4, wherein the software product further establishes a relationship between the contents of the first and second files such that identification of the contents of the first file may be

interpreted by the processor means as identification of the contents of the second file.

6. A method for operating a document processing system in which first and second files are stored, and further in which a relationship has been established between the contents of the first and second files such that selection of the contents of the first file is interpreted by the document processing system as a selection of the contents of the second file, comprising the steps of:
 - designating a processing task to be performed on a document;
 - selecting the contents of the first file for the processing task; and
 - obtaining, by way of selection of the contents of the first file, the processing task being performed on the contents of the second file.

7. A method for operating a data processing resource, comprising the steps of:
 - selecting a document field region on a printed paper form for processing according to a selected process, the paper form of the type having:
 - an identification code known to the data processing resource;
 - an encoded indication of its identification code printed thereon;
 - at least one document field region indicated thereon for receiving a user-imparted indication, which indication being interpreted by the data processing resource as selection of a data representation of a document associated with the document field region, the data representation of the document being stored in a memory apparatus connected to the data processing resource; and
 - a previously assigned arbitrary, image domain document label printed thereon and uniquely associated with a single document field region of the at least one document field regions, for facilitating the selection of the appropriate document field region;
 - inputting the paper form to a facsimile apparatus;
 - causing the facsimile apparatus to convert an image of the paper form, including an image of the selection of the document field region, into an electronic data representation thereof, and transmitting the electronic data representation to the data processing resource;
 - processing the electronic data representation of the printed paper form such that:
 - the encoded indication of the form identification is decoded and the form is identified;
 - from the form identification, the selected document field region is identified, and from this the associated data representation of a document identified; and
 - causing the selected process to be performed on the identified data representation of a document.

8. A method as claimed in claim 7, wherein the selected process is to send the identified data representation of a document to a selected destination, and further comprising the steps of converting the identified data representation of a document into the appropriate format for sending, and sending the converted data representation to the selected destination.

9. A method as claimed in claim 8, wherein the selected destination is a facsimile apparatus, and the step of converting the identified data representation of a document comprises the conversion of the data representation into a standard facsimile protocol format.

10. A method as claimed in claim 7, wherein the selected process is to retrieve the identified data representation of a document to an identified receiving facsimile apparatus, and further comprising the steps of converting the identified data representation of a document into a standard facsimile protocol format, and transmitting the converted data representation to the receiving facsimile apparatus.

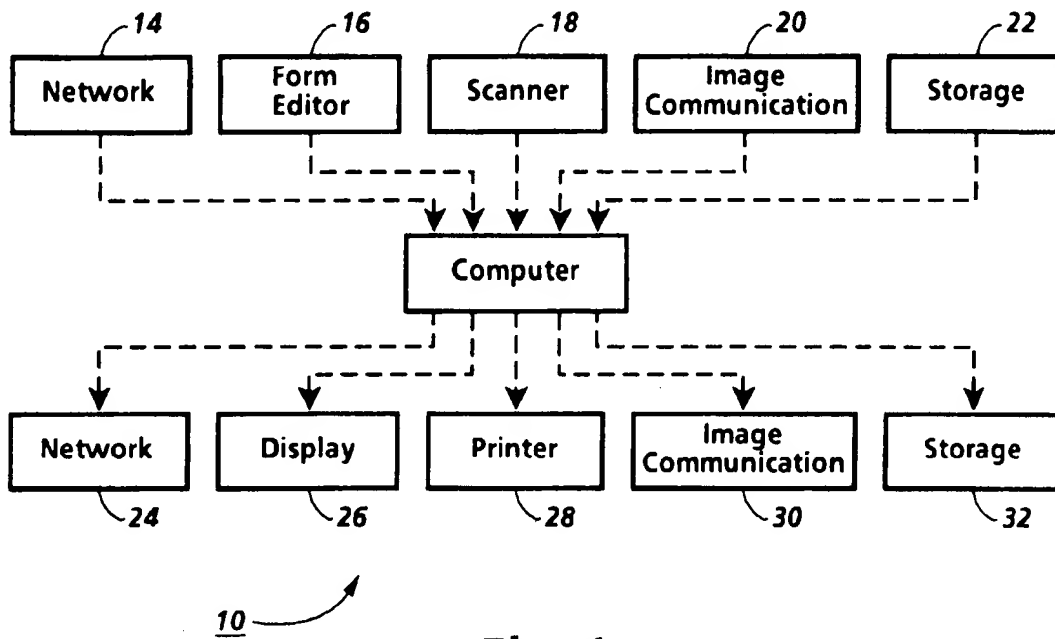


Fig. 1

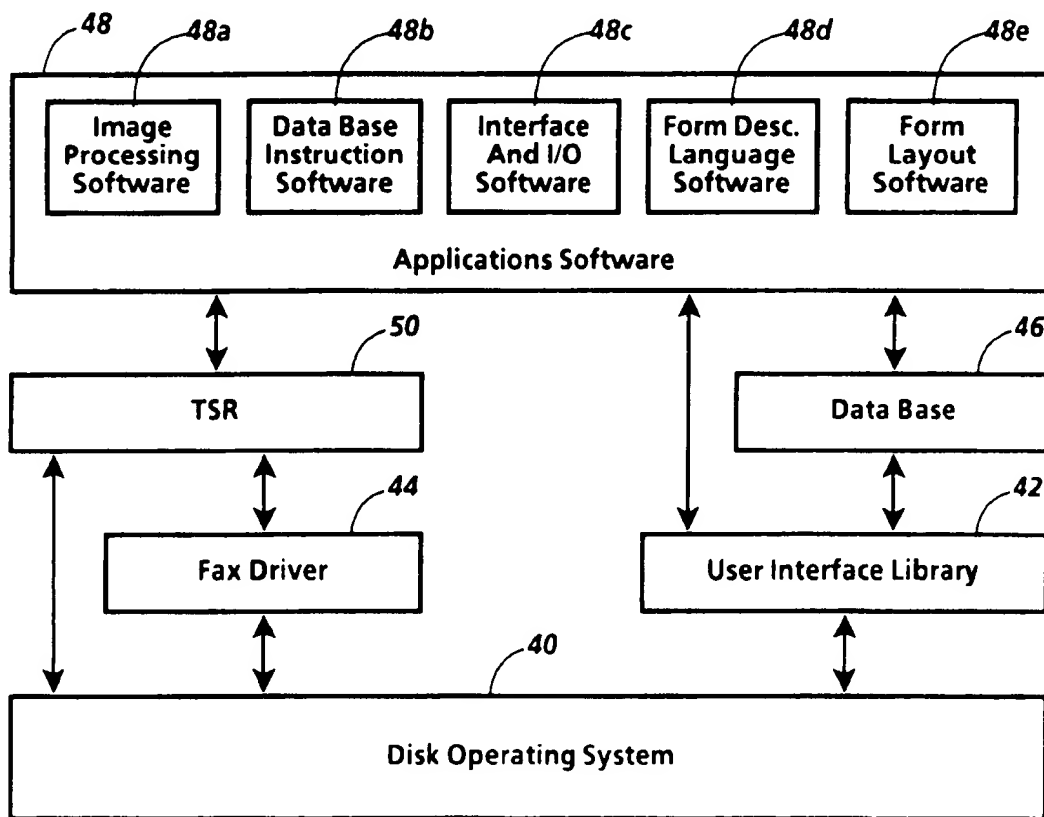
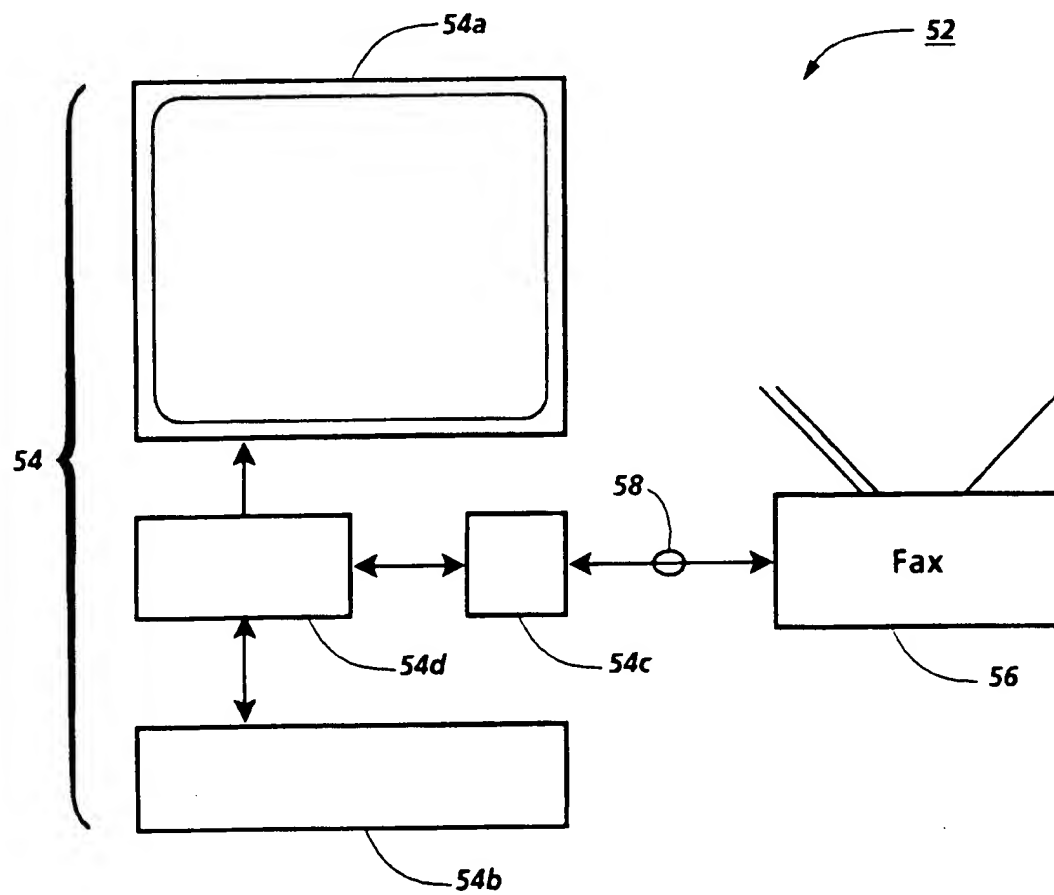


Fig. 2

**Fig. 3**

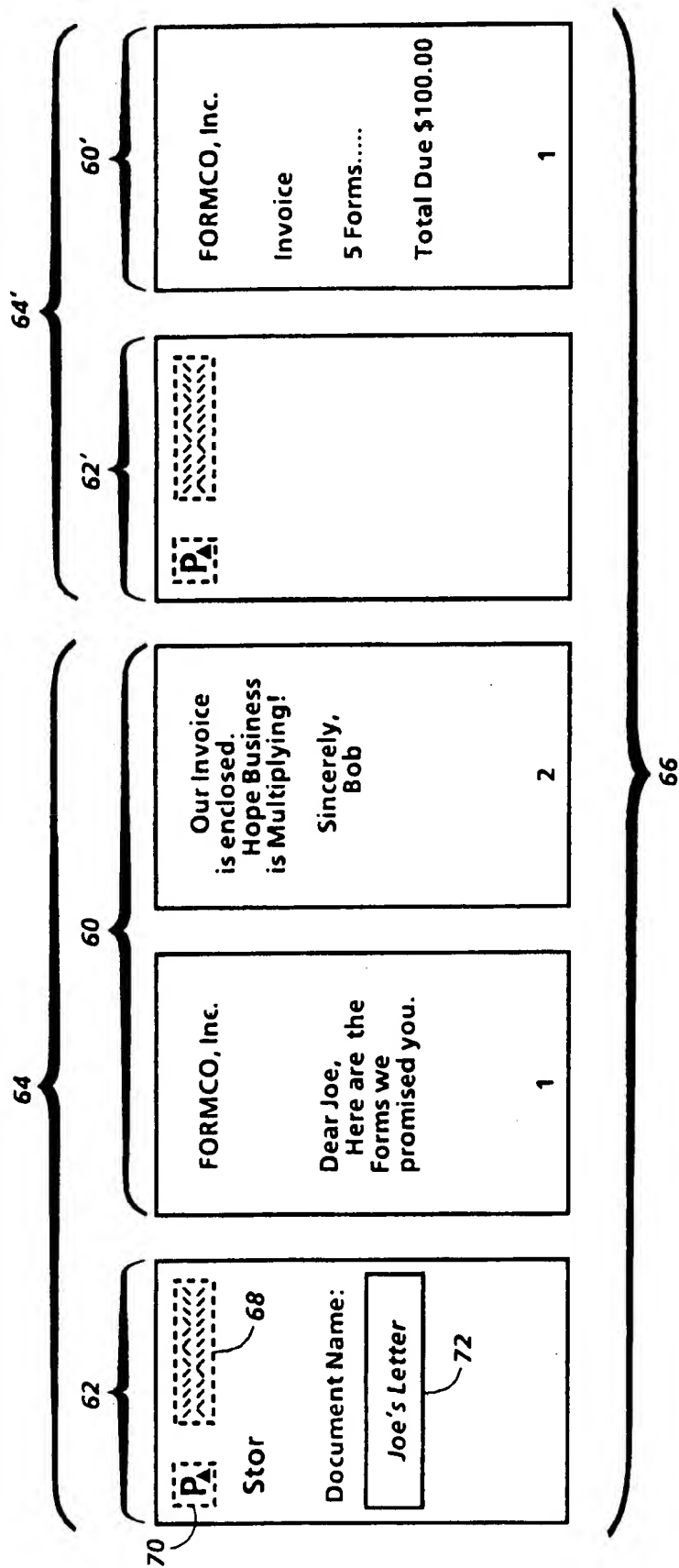
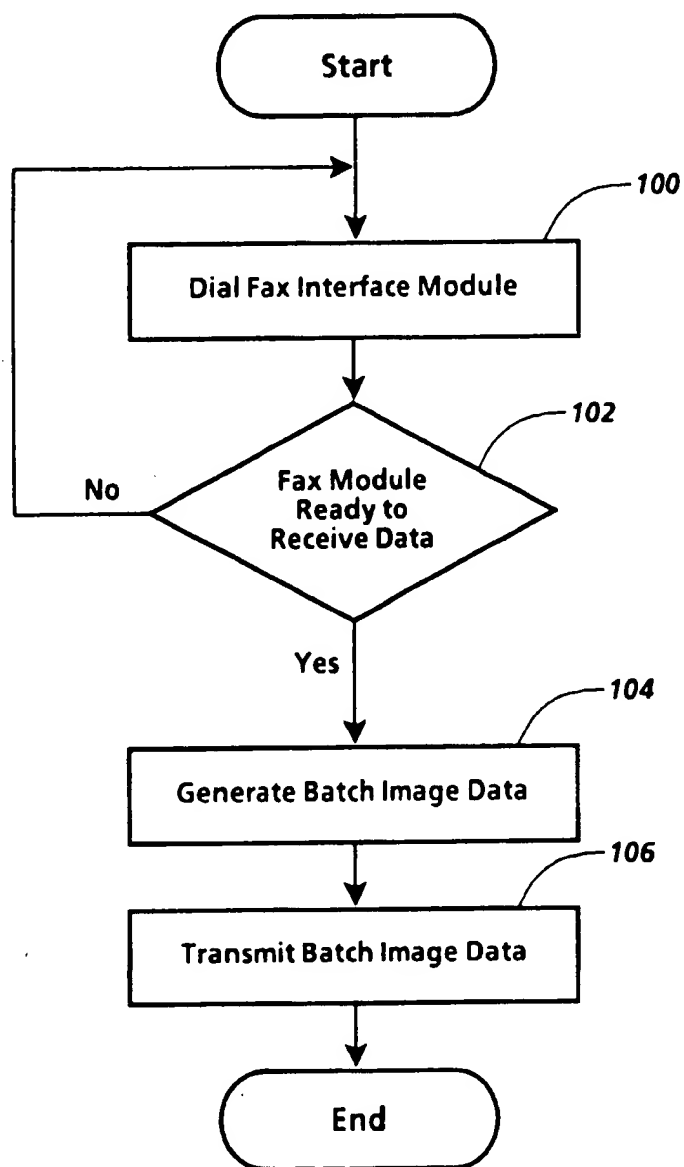
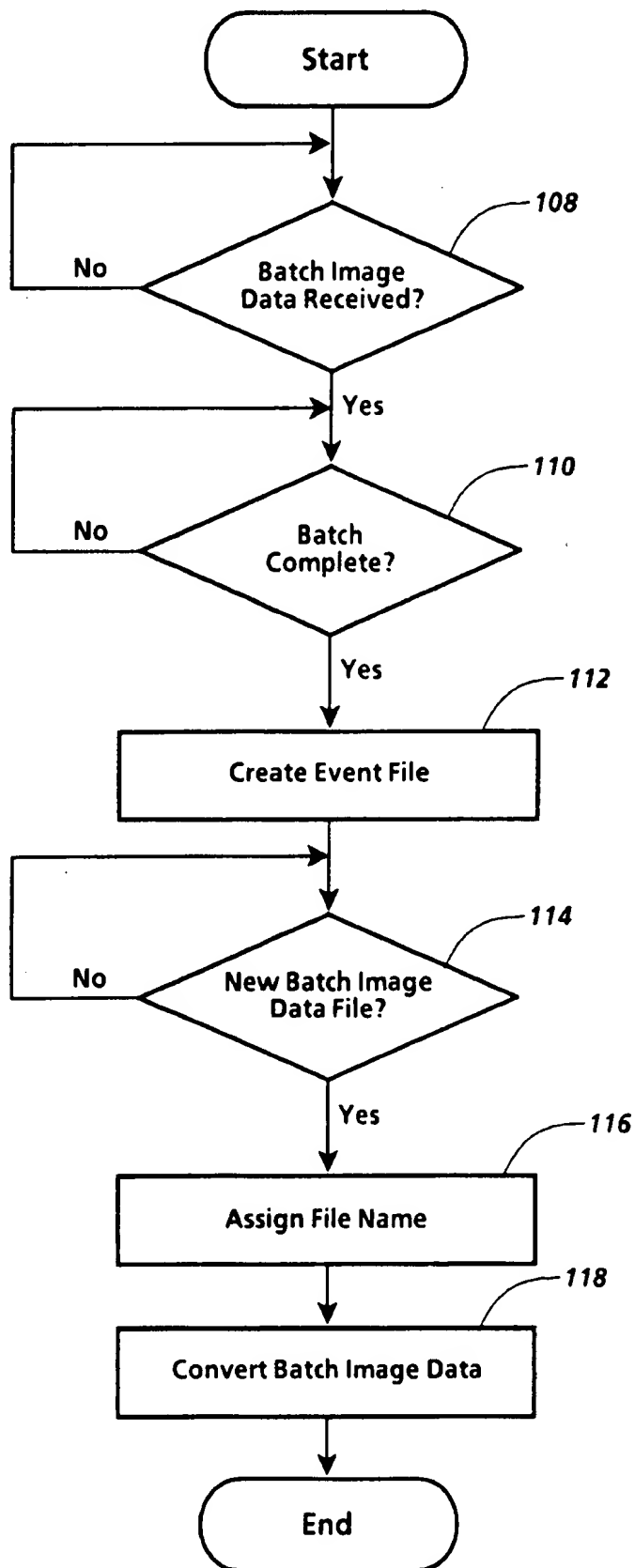
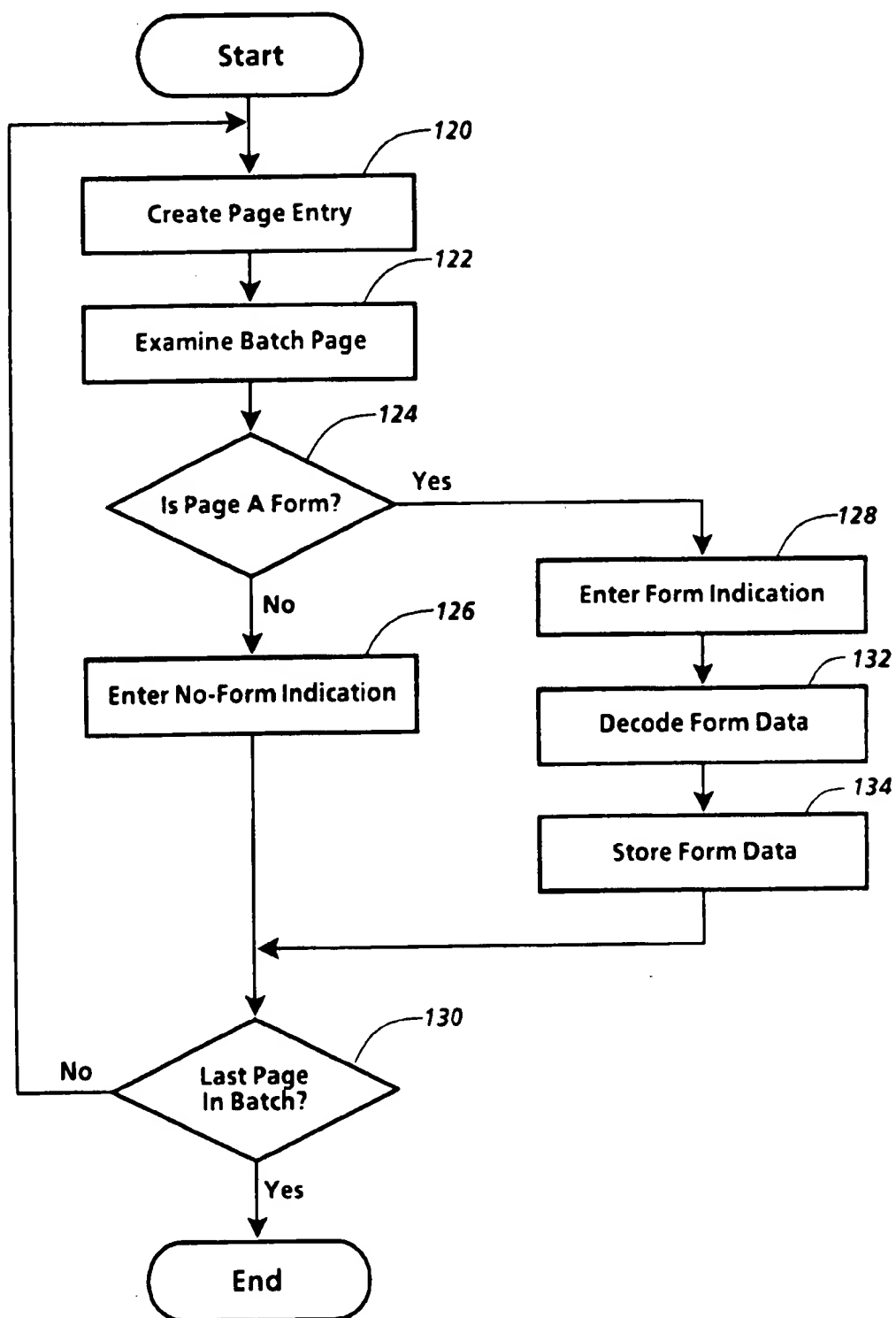
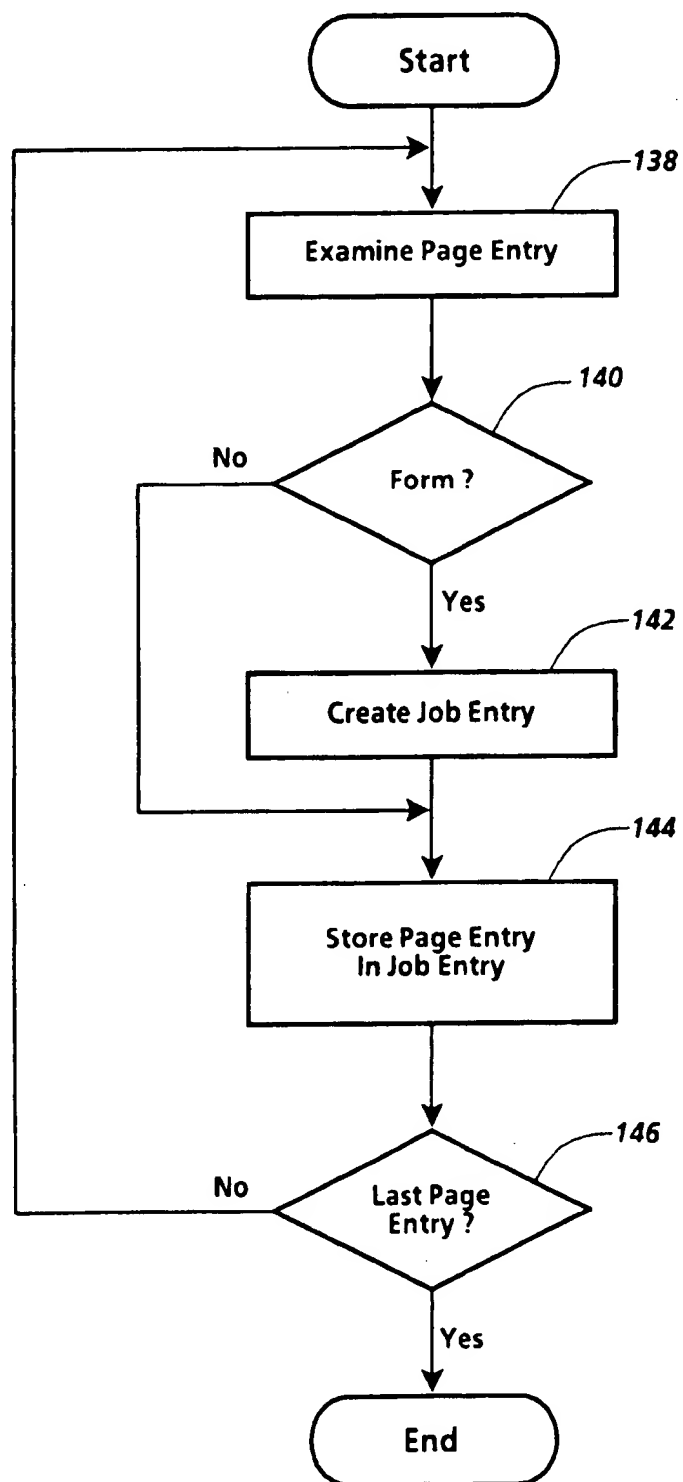


Fig. 4

**Fig. 5**

**Fig. 6**

**Fig. 7**

**Fig. 8**

152 {

154 {

156 {

158 {

Stor

Document Name: 160

Retreive

☐ DOC1 ☐ DOC3

☐ DOC2 ☐ DOC4

List

☐ In Basket ☐ DOC Index

150

Fig. 9

184 Pre-Action Step	186 Modifier	188 State	190 Parameter
Store	—	—	File1.lmg
List	Delete	Y	—
Retrieve	—	Y	FileX.Doc1

170

Fig. 10